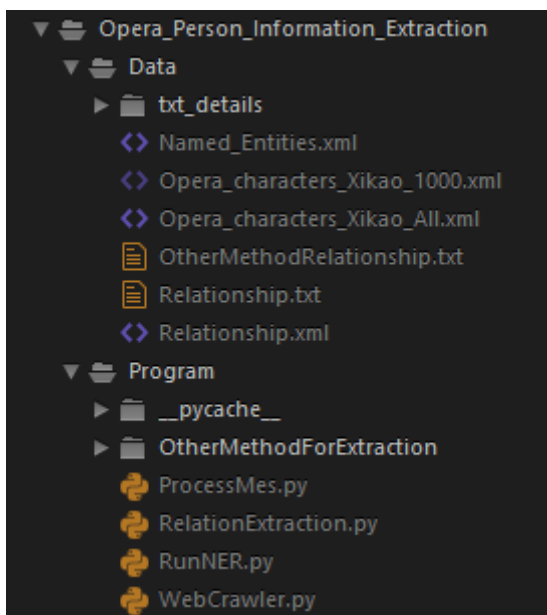

网络爬虫与信息抽取

王瀛 2017213846

1.提交文件内容

提交文件夹包含两部分内容，如下图所示：



其中第一部分是 Data 文件夹，其中包含 txt_details 文件夹，里面是 1000 个京剧人物的详细介绍 txt 文本，每个 txt 文本以该人物的名字命名，正文是一大段详细介绍；Name_Entities.xml 是对于爬到的每一条人物介绍识别出正文部分的人名，地名和戏曲名，以 xml 格式存储；Opera_characters_Xikao_1000.xml 是从爬到的所有京剧人物介绍中提取的前 1000 条数据用于信息抽取；

Opera_characters_Xikao_All.xml 是爬虫爬到的所有京剧人物数据；Relationship.txt 文本是根据正则表达规则抽取出来的实体关系，主要包括师徒，人物和出生地从属关系，人物和科班/学校从属关系；Relationship.xml 则是将 Relationship.txt 中的关系以 xml 格式表示；OtherMethodRelationship.txt 则是利用深度学习的模型抽取出来的人物关系之间的概率分布。

第二部分是 Program 文件夹，包含了本次作业的所有程序。WebCrawler.py 是网络爬虫程序，RunNER.py 是命名实体识别的程序，RelationExtraction.py 是实

体关系抽取的程序，ProcessMes.py 是简单处理 xml 文本的程序；而 OtherMethodForExtraction 文件夹下包含了深度学习方法抽取实体关系的模型，数据和代码。

2.实验介绍

2.1 网络爬虫

代码文件是 WebCrawler.py 文件。

通过分析作业要求中提交中的 5 个网址，发现中国京剧戏考(www.xikao.com)网提供的京剧人物最为充分，包含了 2665 名京剧人物。进入其网页观察发现，首先需要从 4 个子分类（生旦净丑）的网页下爬取所有人物的详细介绍网址，再遍历每个网址去爬虫其中的人物详细信息。

在子分类的网页下，使用如下正则表达式：

```
"<li class=\"bullet\">.*?<a href=\"(.*)\" class.*?>.*?<span class=\"brief_info\">(.*)</span>.*?</li>"
```

爬取人物详细介绍网站，出生逝世年月。之后进入详细介绍网站，爬取京剧人物姓名，性别以及详细信息介绍，利用如下正则表达式：

```
"<div class=\"namecard\">.*?</div>.*?</div>.*?<b>(.*)</b>(.*)<br />(.*)<hr size=\"1\".*?</td>"
```

爬取到的信息以 xml 的格式写入 Opera_characters_Xikao_All.xml 文件，xml 文件的格式如下图所示，

```
<Performers name="安舒元">
  <genre>生行演员</genre>
  <Index>1</Index>
  <Url>http://history.xikao.com/person/安舒元</Url>
  <deathdate>1961</deathdate>
  <name>安舒元</name>
  <sex>男</sex>
  <person_msg>男，京剧老生。北京人。</person_msg>
  <details>12岁从师贵俊卿学老生，14岁以小俊卿艺名在北京城南游艺园与马连良同台演出。20岁时易名安舒元。1924年后，曾先后四次到上海演出，并到武汉、天津等地献艺。20世纪40年代与田汉结识。1943年与黄玉麟、白家麟、高百岁、王少楼、徐兰沅等在北京合作组班，排演了《封神榜》、《八仙得道》、《狸猫换太子》、《水浒传》等连台本戏。1949年后，经田汉推荐到戏曲实验学校（后名中国戏曲学校）任教，学生有李鸣岩、金桐、陈国卿等。1958年调吉林省戏曲学校任教。1961年在长春病逝。安舒元早年嗓音有些左，调门高，经不断研磨，唱腔颇有韵味，他以唱念讲究、表演细腻著称，且文武全能，时与贯大元、王文源并称“老生三元”。曾灌制《乌龙院》（与筱翠花合作）、《四郎探母》、《定军山》等唱片存世。</details>
  <birthdate>1900</birthdate>
</Performers>
```

其中<genre>表示该京剧演员的行当（角色类型），<Index>是序号，<Url>是该京剧人物详细介绍的网站地址，<deathdate>是该人物的逝世日期（部分人物没有），<name>是人物姓名，<sex>是性别，<person_msg>是人物的简单介绍，<details>是人物的详细介绍，<birthdate>是人物的出生日期。以上就是一个京剧人物爬取到的所有信息。获取其中前 1000 条进行后面的实体分析。

2.2 命名实体识别

代码文件是 RunNER.py 文件。

使用 Stanford NLP 包进行姓名和地点的命名实体识别，而戏曲名则使用书名号“《》”进行正则匹配。

在进行命名实体识别之前，首先调用 ProcessMes.py 中的 getDetailMes 函数将 xml 中爬取到的 1000 条人物介绍以迭代器的方法返回。

命名实体识别使用 Stanford 的 NLP 工具包，首先在 Stanford NLP 官网（<https://stanfordnlp.github.io/CoreNLP/index.html>）下载最新的模型文件，包括：

（1）CoreNLP 完整包

（<http://nlp.stanford.edu/software/stanford-corenlp-full-2017-06-09.zip>）：下载后解压到工具目录；

（2）中文模型

（<http://nlp.stanford.edu/software/stanford-chinese-corenlp-2017-06-09-models.jar>）：下载后复制到上述工作目录。

由于本项目使用 python 语言进行编写，而 Stanford NLP 是使用 java 语言开发，因此还需要安装 stanza（<https://github.com/stanfordnlp/stanza>）的 python 包，用于在 python 环境下调用 Stanford NLP 工具。可以在上述 github 网站上下载后安装，也可以使用命令“pip install stanza”直接进行下载。

下载好上述所有工具后，在 python 程序中使用 NLP 函数之前，需要在 NLP 工作目录下启动 cmd 窗口输入如下命令启动本地服务器：

```
java -mx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 9000 -timeout 15000
```

服务器启动后,在 python 代码中导入 stanza.nlp.corenlp 包中的 CoreNLPClient,之后即可使用如下代码

```
client = CoreNLPClient(server='http://localhost:9000', default_annotators=['ssplit', 'lemma', 'tokenize', 'pos', 'ner'])
```

在 Python 中创建一个 CoreNLPClient 对象,并调用该对象相应的函数进行命名实体识别。

首先进行戏曲名的识别,戏曲名识别使用书名号“《》”进行正则匹配"(《.*?》)",找到正文中所有戏曲名后,存入 dict 中,为了防止之后识别姓名和地名出现戏曲名中的姓名或者地方,存储戏曲名后使用“#”将其替代。姓名和地名识别代码如下:

```
annotated = client.annotate(details)
for sentence in annotated.sentences:
    for token in sentence:
        if token.ner == "PERSON":
            if token.word not in name_list and len(token.word) != 1:
                name_list.append(token.word)
        if token.ner == "GPE":
            if token.word not in location_list and len(token.word) != 1:
                location_list.append(token.word)
```

循环每一个句子,判断某个句子中的词的 Ner 标注是否属于“PERSON”或者“GPE”,其中需要把词长度等于 1 的情况去除(与实际不相符)。这样就将人物正文介绍中出现的姓名和地点进行了筛选。

识别出姓名,地名以及戏曲名后,将其以 xml 格式存入,存入实例如下:

```

<Performers name="安舒元">
  <Index>1</Index>
  <name>安舒元</name>
  <name_list>
    <name_list_value>安舒元</name_list_value>
    <name_list_value>马连良</name_list_value>
    <name_list_value>田汉</name_list_value>
    <name_list_value>白家麟</name_list_value>
    <name_list_value>王少楼</name_list_value>
    <name_list_value>徐兰沅</name_list_value>
    <name_list_value>李鸣岩</name_list_value>
    <name_list_value>陈国卿</name_list_value>
    <name_list_value>王文源</name_list_value>
  </name_list>
  <location_list>
    <location_list_value>北京</location_list_value>
    <location_list_value>上海</location_list_value>
    <location_list_value>武汉</location_list_value>
    <location_list_value>天津</location_list_value>
    <location_list_value>吉林省</location_list_value>
    <location_list_value>长春</location_list_value>
  </location_list>
  <operaname_list>
    <operaname_list_value>《封神榜》</operaname_list_value>
    <operaname_list_value>《八仙得道》</operaname_list_value>
    <operaname_list_value>《狸猫换太子》</operaname_list_value>
    <operaname_list_value>《水浒传》</operaname_list_value>
    <operaname_list_value>《乌龙院》</operaname_list_value>
    <operaname_list_value>《四郎探母》</operaname_list_value>
    <operaname_list_value>《定军山》</operaname_list_value>
  </operaname_list>
</Performers>

```

2.3 实体关系抽取

代码文件为 RelationExtraction.py 文件。代码测试时输入需要进行测试的文件夹路径以及输出结果的保存路径和文件名。输入空格则表示默认输入路径为“../Data/txt_details”，输出结果文件为“../Data/Relationship.txt”。

实体关系抽取使用了正则表达规则匹配的方法进行实体关系的抽取，通过观察师徒，出生地以及科班/学校的文本表达特性，总结出相应的规则，并将规则以正则表达式的方式去文本中匹配。

由于需要识别出句子中的实体类别，分句以及分词，故还需调用第二步介绍的 NLP 工具包，在 NLP 工作目录下的 cmd 中输入 `java -mx4g -cp "edu.stanford.nlp.pipeline.StanfordCoreNLPServer -port 9000 -timeout 15000` 命令打开本地服务器，之后在 python 代码中调用相应函数即可。

循环正文中的每一个句子，判断是否匹配师徒关系正则表达，若匹配则找到句子中的“PERSON”标注。由于师徒关系的主语不同，故使用两个正则表达去

匹配师徒关系，首先是抽取该名家的师父，正则表达式如下："(.*师从)|(.*(从|向).*学.)|(.*(拜.*))|(.*(被.*收为.*))|(.*(从师.*))|(.*(开蒙(?!戏).*))|(.*(师承.*))|(.*(为师.*))|(.*(受.*指导.*))|(.*(被.*收为.*))|(.*(为.*弟子))|(.*(得.*真传.*))|(.*(传授.*))|(.*(亲授.*))|(入.*习)"，而抽取改名家徒弟的正则表达式如下："(.*培养.*)|(.*(收.*弟子.*))"。

人物和出生地从属关系以及人物和科班/学校从属关系同师徒关系的处理，使用的正则表达关系分别如下："(.*(?<!族)人)|(.*(生于.*))"、"(?(?!调)(入|从|在|到|考入|毕业于|坐科)(.*?(?<!(的)))(学校|科班|学院)(?!任教)"。处理完这 3 种关系后将关系写入 Relationship.txt 和 Relationship.xml 文件中，例子如下：

```
杨长秀----->
米福生 师徒 杨长秀
吴泽东 师徒 杨长秀
李文才 师徒 杨长秀
刘福生 师徒 杨长秀
马玉璋 师徒 杨长秀
丁晨元 出生地 北京
丁晨元 习艺 中国戏曲学院
杨长秀 习艺 学院
```

```
<Performers name="丁晨元">
  <Master>
    <Master_relationship>杨长秀 师徒 丁晨元</Master_relationship>
    <Master_relationship>米福生 师徒 丁晨元</Master_relationship>
    <Master_relationship>吴泽东 师徒 丁晨元</Master_relationship>
    <Master_relationship>李文才 师徒 丁晨元</Master_relationship>
    <Master_relationship>刘福生 师徒 丁晨元</Master_relationship>
    <Master_relationship>马玉璋 师徒 丁晨元</Master_relationship>
  </Master>
  <Location>
    <Location_relationship>丁晨元 出生地 北京</Location_relationship>
  </Location>
  <School>
    <School_relationship>丁晨元 习艺 中国戏曲学院</School_relationship>
    <School_relationship>丁晨元 习艺 学院</School_relationship>
  </School>
</Performers>
```

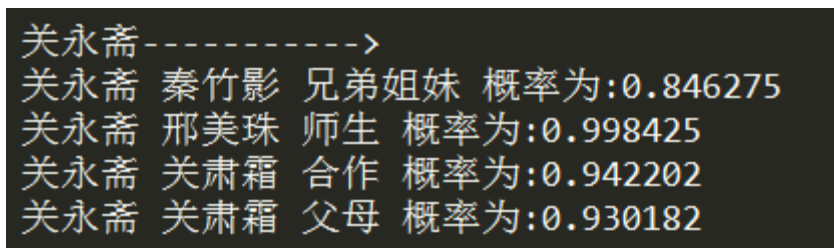
2.4 额外加分（关系抽取）

对于第三步，在 [github](#) 上找到一个使用 Bi-GRU 和字向量做端到端的中文关系抽取的项目 RE_BGRU_2ATT，链接如下：

https://github.com/crownpku/Information-Extraction-Chinese/tree/master/RE_BGRU_2ATT。利用该项目训练好的模型，在 1000 个文本上抽取出概率大于 0.8 的 10 种关系，分别为父母，夫妻，师生，兄弟姐妹，合作，情侣，祖孙，好友，亲戚，同门以及上下级。

代码，数据以及模型位于 Program 文件夹下的 OtherMethodForExtraction 文件夹。测试时输入需要进行测试的文件夹路径以及输出结果的保存路径和文件名。输入表格表示默认输入路径为“../Data/txt_details”，输出结果文件为“../Data/OtherMethodRelationship.txt”。

RE_BGRU_2ATT 项目使用双向 GRU，字与句子的双重 Attention 模型，输入两个姓名实体和一个句子，从句子中判断该两个实体属于以上 10 种关系的概率。首先遍历文本中的所有句子，若存在标注为“PERSON”的实体，则将正文的各家姓名，该实体姓名以及句子放入 RE_BGRU_2ATT 的模型中进行关系抽取，输出为概率大于 0.8 的 10 种关系之一。抽取的关系如下图所示：



```
关永斋----->
关永斋 秦竹影 兄弟姐妹 概率为:0.846275
关永斋 邢美珠 师生 概率为:0.998425
关永斋 关肃霜 合作 概率为:0.942202
关永斋 关肃霜 父母 概率为:0.930182
```

2.5 心得体会

这次作业使用爬虫爬取数据并对数据进行实体分析让我收获了很多，对如何获取数据，以及如何对数据进行分析有了更深的了解。

在短时间内学习并编写爬虫程序，Stanford NLP 工具的使用以及深度学习关系抽取模型的使用都使我受益匪浅，总之，这次作业虽然花了我比较多的时间，但是最后的结果还是比较满意。